

An algorithm to perform POVMs through Neumark theorem: application to the discrimination of non-orthogonal pure quantum states

Wilson R. M. Rabelo,^{*} Alexandre G. Rodrigues, and Reinaldo O. Vianna[†]

*Grupo Informação Quântica - Departamento de Física - ICEX - CP 702 -
Universidade Federal de Minas Gerais - 30123-970 - Belo Horizonte - MG - Brazil*

(Dated: 9th February 2008)

We consider a protocol to perform the optimal quantum state discrimination of N linearly independent non-orthogonal pure quantum states and present a computational code. Through the extension of the original Hilbert space, it is possible to perform an unitary operation yielding a final configuration, which gives the best discrimination without ambiguity by means of von Neumann measurements. Our goal is to introduce a detailed general mathematical procedure to realize this task by means of semidefinite programming and norm minimization. The former is used to fix which is the best detection probability amplitude for each state of the ensemble. The latter determines the matrix which leads the states to the final configuration. In a final step, we decompose the unitary transformation in a sequence of two-level rotation matrices.

PACS numbers:

I. INTRODUCTION

Quantum state discrimination is a crucial problem in quantum information theory, and it is well known the impossibility of doing this perfectly for non-orthogonal states [1]. Nevertheless, if a non zero probability of inconclusive results is allowed for, it is possible to never mistake a state for another, by means of an appropriate Positive Operator Valued Measure (POVM). This strategy is known as unambiguous state discrimination (USD), and the best procedure of this kind is that which minimizes the probability of inconclusive results. The discrimination of two equally probable non-orthogonal pure states was firstly considered by Ivanovich [3], Dieks [4] and Peres [5]. The case of two states with unequal prior probabilities was treated by Jaeger and Shimony [6]. Chefles [7] showed that USD of N pure quantum states is possible if and only if they are linearly independent.

With the aid of Semidefinite Programming (SDP), Eldar [8] obtained the set of necessary and sufficient conditions the USD measurement operators satisfy. She also showed that for a given set of pure states, there always is an ensemble for which the optimal USD detects each state with equal probability (i.e, an Equal Probability Measurement - EPM).

In references [10] and [11], the USD problem is approached via the Neumark theorem [1], i.e, the realization of a POVM by means of projective measurements in an extended Hilbert space. In [10], it is done in the context of linear quantum optics, whereas in

[11] an ions trap architecture is considered.

Our main goal is to derive the transformation which maps N non-orthogonal pure states in a set of states that can be discriminated by usual projective measurements in an extended Hilbert space. It is equivalent to a generalized measurement in the original space, and is the content of Neumark's theorem. We then present a simple MATLAB code that takes as input the ensemble of non-orthogonal states and outputs the best set of discriminable states and the pertinent transformations. We believe this computational code can be pedagogically useful to anyone getting started with quantum information and semidefinite programming.

The paper is organized as follows. In section I we discuss generalized measurements and semidefinite programming. In section II we present a protocol to discriminate pure non-orthogonal states and, in section IV, we exercise with a numerical example in the context of quantum key distribution. We conclude in section V. In the appendix, we furnish our MATLAB code.

II. POVMS AND DISCRIMINATION WITH SDP

The operation commonly known as positive operator-valued measure can be performed with a set of quantum detection operators Π_k , where the probability p_k of obtaining the state labeled by the index k is given by

$$p_k = \text{Tr}(\Pi_k \rho), \quad (1)$$

where ρ is the density operator of the system. As the probabilities are obviously non-negative reals and sum to one, all the quantum detection operators are

^{*}Electronic address: wilson@unifap.br

[†]Electronic address: reinaldo@fisica.ufmg.br

semidefinite operators and form a resolution of the identity,

$$\sum_k \Pi_k = I. \quad (2)$$

The problem of USD for N pure non-orthogonal states can be stated as follows. We assume that a quantum system was prepared in one of the N pure states $\{|Q_i\rangle\}$, $i = 1, \dots, N$. Each state $|Q_i\rangle$ is in a N -dimensional Hilbert space. In order to identify a state or to return an inconclusive result, the measurement operators must obey $\langle Q_i | \Pi_k | Q_i \rangle = p_i \delta_{ik}$ with $0 \leq p_i \leq 1$. Therefore, to satisfy condition (2), we have $\sum_{i=1}^N \Pi_i = I$ and the inconclusive result is given by $\Pi_0 = I - \sum_{i=1}^N \Pi_i$.

In [8], Eldar showed that the optimal unambiguous discrimination can be formulated as a SDP problem. The measurement operators are expressed in the form

$$\Pi_i = p_i |\tilde{Q}_i\rangle\langle\tilde{Q}_i|, \quad (3)$$

where each state $|\tilde{Q}_i\rangle$ is in a N -dimensional Hilbert space, and the Π_i are not orthogonal projectors in this space. The vectors $|\tilde{Q}_i\rangle$ are the *reciprocal states* associated with $|Q_i\rangle$, such that

$$\langle\tilde{Q}_i|Q_k\rangle = \zeta_{ik} \delta_{ik}, \quad 1 \leq i, k \leq N \quad (4)$$

where the factor ζ_{ik} indicates the scalar product is not normalized. Given the matrix Ψ , whose columns are the vectors $|Q_i\rangle$, the $|\tilde{Q}_i\rangle$ are the columns of the matrix $\tilde{\Psi}$, namely,

$$\tilde{\Psi} = \Psi(\Psi^* \Psi)^{-1}. \quad (5)$$

Given an ensemble ρ , where each state $|Q_i\rangle$ is prepared with probability μ_i , the total probability of a successful detection is

$$P_D = \sum_{i=1}^N \mu_i \langle Q_i | \Pi_i | Q_i \rangle = \sum_{i=1}^N \mu_i p_i. \quad (6)$$

The problem of optimal USD is then to find measurement operators Π_i , or equivalently, the probabilities p_i , which maximize P_D , subject to the constraint (2), which can be recast as

$$I - \sum_{i=1}^N p_i |\tilde{Q}_i\rangle\langle\tilde{Q}_i| \geq 0. \quad (7)$$

A SDP problem is to find $x \in R^m$, which minimizes the linear function $c^T x$, subject to the matrix inequality $F(x) = F_0 + \sum_{i=1}^m x_i F_i \geq 0$, where the problem data are the vector $c \in R^m$ and the $m+1$ complex

Hermitian matrices F_i [12]. It is known as the primal formulation of SDP.

Equations (6) and (7) can be recast as a SDP problem, namely:

$$\min_{p \in R^N} \{-\mu^T p\}, \quad (8)$$

subject to $N+1$ constraints,

$$\begin{aligned} I - \sum_{i=1}^N p_i |\tilde{Q}_i\rangle\langle\tilde{Q}_i| &\geq 0, \\ p_i &\geq 0, \quad 1 \leq i \leq N. \end{aligned} \quad (9)$$

III. PROTOCOL FOR OPTIMAL DISCRIMINATION OF PURE STATES VIA ROTATIONS AND VON NEUMANN MEASUREMENTS

We start out by rewriting the N entry states to be discriminated in a ladder form in the orthonormal basis $\{|i\rangle, i = 1, \dots, N\}$,

$$\begin{aligned} |Q_1\rangle &= |1\rangle, \\ |Q_2\rangle &= c_{12}|1\rangle + c_{22}|2\rangle, \\ |Q_3\rangle &= c_{13}|1\rangle + c_{23}|2\rangle + c_{33}|3\rangle \\ &\vdots \\ |Q_N\rangle &= c_{1N}|1\rangle + \dots + c_{NN}|N\rangle, \end{aligned} \quad (10)$$

where $\{|Q_i\rangle\} \in \mathcal{H}_N$ (N -dimensional Hilbert space). This can always be done by means of a unitary transformation U_0 . Then, to apply Neumark's theorem, we extend the original Hilbert space, e.g. through addition of ancillas, to $2N-1$ dimensions and map the original states to the *final configuration*,

$$\begin{aligned} |Q_{1f}\rangle &= g_1|1\rangle + g_{N+1}|N+1\rangle + \dots + g_{2N-1}|2N-1\rangle \\ |Q_{2f}\rangle &= g_2|2\rangle + g_{2N}|N+1\rangle + \dots + g_{3N-2}|2N-1\rangle \\ |Q_{3f}\rangle &= g_3|3\rangle + g_{3N-1}|N+1\rangle + \dots + g_{4N-4}|2N-2\rangle \\ &\vdots \\ |Q_{if}\rangle &= g_i|i\rangle + g_{[\frac{i}{2}(2N+3-i)-1]}|N+1\rangle + \\ &\quad \dots + g_{[N(i+1)+\frac{i}{2}(1-i)-1]}|2N+1-i\rangle \\ &\vdots \\ |Q_{Nf}\rangle &= g_N|N\rangle + g_{[\frac{1}{2}N(N+3)-1]}|N+1\rangle. \end{aligned} \quad (11)$$

Now, a projective measurement in the orthonormal basis $\{|i\rangle, i = 1, \dots, 2N-1\}$ yields an unambiguous discrimination. The state labeled by i is identified with probability g_i^2 , when the measurement collapses to $|i\rangle$ for $1 \leq i \leq N$. For other values of i the result is inconclusive. Therefore, the first N g_i are chosen to

produce the best possible discrimination, i.e, $g_i = \sqrt{p_i}$ for the p_i defined in Eq.3, which are determined by SDP. The other coefficients $\{g_i, i = N+1 \text{ to } \frac{N}{2}(N+3) - 1\}$ are fixed in order to preserve normalization and the scalar products among the original states.

Once the final configuration is known, we want to determine the unitary transformation U_1 which maps the original states to it. Norm minimization [14] implies,

$$\sum_{i=1}^N \||Q_{if}\rangle - U_1|Q_i\rangle\|^2 = 2N - 2\text{Re} \left[\text{Tr}(A'U_1^\dagger) \right] = 0, \quad (12)$$

where

$$A' = \sum_{i=1}^N |Q_{if}\rangle\langle Q_i|. \quad (13)$$

A' is a singular $(2N-1) \times (2N-1)$ complex matrix. The singular value decomposition (SVD) of A' is $A' = V\Sigma W^\dagger$, with V and W $(2N-1) \times (2N-1)$ unitary matrices, and Σ a diagonal $(2N-1) \times (2N-1)$ matrix, then U_1 is given by

$$U_1 = VW^\dagger. \quad (14)$$

The unitary transformation $U = U_1U_0$ can now be decomposed [15] in a sequence of rotations (R_{kl}) in the hyperplanes (kl), as

$$U = \prod_{k=1}^{N-1} \prod_{l=k+1}^N R_{kl}^\dagger. \quad (15)$$

The R_{kl} are two-level matrices, with the four non trivial entries,

$$[R_{kl}]_{kk} = \frac{[U]_{kk}^*}{\sqrt{|[U]_{kk}|^2 + |[U]_{lk}|^2}},$$

$$[R_{kl}]_{ll} = -[R_{kl}]_{kk}^*,$$

$$[R_{kl}]_{kl} = \frac{[U]_{lk}}{\sqrt{|[U]_{kk}|^2 + |[U]_{lk}|^2}},$$

$$[R_{kl}]_{lk} = [R_{kl}]_{kl}^*.$$

The trivial entries are

$$[R_{kl}]_{mm} = 1, m \neq (k, l),$$

$$[R_{kl}]_{mn} = 0, (m, n) \neq (k, l).$$

We can also express the rotations R_{kl}^\dagger in terms of Pauli matrices, $R_z^k(\theta) = \exp(-i\theta\sigma_z/2)$ (the super-index k indicates the rotation is over the ket $|k\rangle$), $R_y^l(\theta) = \exp(-i\theta\sigma_y/2)$. Hence,

$$R_{kl}^\dagger(\alpha, \beta, \gamma, \delta) = e^{i\alpha} R_z^k(\beta) R_y^l(\gamma) R_z^k(\delta). \quad (16)$$

Therefore the unitary transformation $U = U_1U_0$, followed by a projective measurement in the basis $\{|i\rangle, i = 1, \dots, 2N-1\}$, discriminates unambiguously any N pure non-orthogonal states. Summarizing, we have the following algorithm:

- Rewrite the entry states in a ladder form (U_0).
- Fix the *conclusive amplitudes* (g_1 to g_N) using SDP.
- Fix the *inconclusive amplitudes* (g_{N+1} to $g_{\frac{N}{2}(N+3)-1}$) such that normalization and scalar products among the states be preserved.
- Build the unitary transformation (U_1) that maps the entry states in the ladder form to the *final discriminable configuration*.
- Decompose $U = U_1U_0$ as a product of one-qudit rotations.

IV. EXAMPLE

As an example of application of our code, we consider how a spy (Eve) could use USD to eavesdrop [16, 17] two parties (Alice and Bob) establishing a BB84 [18] cryptographic key.

Alice and Bob are communicating by means of low intensity laser pulses in a network of lossy optical fibers. The signals sent by Alice are coherent states with a certain mean number of photons (μ). The polarization of the pulses are randomly chosen with equal probability among four possibilities, namely, diagonal to right or left ($|d+\rangle$, $|d-\rangle$), and circularly polarized to right or left ($|c-\rangle$, $|c+\rangle$). Bob directs the pulses he receives to detectors, which do not distinguish photon numbers, preceded by polarization analyzers which can be either for linear or circular polarization, also chosen randomly with equal probability. After a certain number of pulses, Alice publicly announces the sequence of polarization basis she used. Bob then checks which pulses he detected using the compatible polarization analyzer. After discounting dark counts and losses in the fibers, the matching detections allow Alice and Bob to establish a cryptographic key, i.e., a long sequence of zeroes and ones (say $|d+\rangle$, $|c+\rangle$ for one and $|d-\rangle$, $|c-\rangle$ for zero). This is the very well known BB84 protocol.

Eve could try to obtain this key as follows. She probes the network and measures the mean number of photons in the pulses, which can be done without disturbing the polarization. Now she knows that most of the time Bob receives states of the type $|polarization\rangle^{\otimes\mu}$. Suppose μ is 3. Eve then prepares an USD scheme for the states $|d+\rangle|d+\rangle|d+\rangle$, $|d-\rangle|d-\rangle|d-\rangle$, $|c+\rangle|c+\rangle|c+\rangle$, $|c-\rangle|c-\rangle|c-\rangle$.

Running our code, all the pertinent parameters to prepare the USD are yielded. In particular, we learn that Eve correctly identifies the polarization state with a probability of 50%, if the state has three photons. When she succeeds, she prepares a state with

the correct polarization and sends it to Bob, who will never know it came from Eve. If she fails, she does nothing at all, and Bob could think it is a dark count or a loss in the network.

$$U_0 = \begin{bmatrix} 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 & 0.3536 \\ 0.3536 & -0.3536 & -0.3536 & 0.3536 & -0.3536 & 0.3536 & 0.3536 & -0.3536 \\ 0.6124 & -0.2041i & -0.2041i & -0.2041 & -0.2041i & -0.2041 & -0.2041 & 0.6124i \\ 0.6124 & 0.2041i & 0.2041i & -0.2041 & 0.2041i & -0.2041 & -0.2041 & -0.6124i \\ 0 & -0.8165 & 0.4082 & 0 & 0.4082 & 0 & 0.0000 & 0 \\ 0 & 0 & -0.1736 & -0.5414 & 0.1736 & 0.7707 & -0.2293 & 0 \\ 0 & 0 & -0.1736 & -0.5414 & 0.1736 & -0.2293 & 0.7707 & 0 \\ 0 & 0 & 0.6631 & -0.2836 & -0.6631 & 0.1418 & 0.1418 & 0 \end{bmatrix}.$$

Let's consider the four quantum states

$$|Q_1^i\rangle = |d+\rangle|d+\rangle|d+\rangle \quad (17)$$

$$|Q_2^i\rangle = |d-\rangle|d-\rangle|d-\rangle \quad (18)$$

$$|Q_3^i\rangle = |c+\rangle|c+\rangle|c+\rangle \quad (19)$$

$$|Q_4^i\rangle = |c-\rangle|c-\rangle|c-\rangle, \quad (20)$$

where

$$|d+\rangle = (|0\rangle + |1\rangle)/\sqrt{2}$$

$$|d-\rangle = (|0\rangle - |1\rangle)/\sqrt{2}$$

$$|c+\rangle = (|0\rangle + i|1\rangle)/\sqrt{2}$$

$$|c-\rangle = (|0\rangle - i|1\rangle)/\sqrt{2}.$$

Rewrite the entry states in a ladder form

$$\begin{aligned} |Q_1\rangle &= |1\rangle \\ |Q_2\rangle &= |2\rangle \\ |Q_3\rangle &= -(0.25 - 0.25i)|1\rangle + \\ &\quad -(0.25 + 0.25i)|2\rangle + 0.8660|3\rangle \\ |Q_4\rangle &= -(0.25 + 0.25i)|1\rangle + \\ &\quad -(0.25 - 0.25i)|2\rangle + 0.8660|4\rangle, \end{aligned} \quad (21)$$

where the unitary matrix U_0 for the transformation is given above.

The best conclusive probability amplitude given by the SDP technique is given by $g_1=g_2=g_3=g_4=0.7071$, where the weight of each state in the ensemble is $\mu_1=\mu_2=\mu_3=\mu_4=0.25$. The inconclusive amplitudes (g_5 to g_{13}) are fixed such that normalization and scalar products among the states be preserved. For this case, $g_5 = -0.3536 + 0.3536i$, $g_6 = -0.3536 - 0.3536i$, $g_7 = 0$, $g_8 = -0.3536 - 0.3536i$, $g_9 = -0.3536 + 0.3536i$, $g_{10} = 0$, $g_{11} = 0$, $g_{12} = 0.7071$ and $g_{13} = 0.7071$.

The final discriminable configuration in the extended Hilbert space is

$$\begin{aligned} |Q_{1f}\rangle &= 0.7071|1\rangle - (0.3536 - 0.3536i)|5\rangle \\ &\quad -(0.3536 + 0.3536i)|6\rangle \\ |Q_{2f}\rangle &= 0.7071|2\rangle - (0.3536 + 0.3536i)|5\rangle \\ &\quad -(0.3536 - 0.3536i)|6\rangle \\ |Q_{3f}\rangle &= 0.7071|3\rangle + 0.7071|6\rangle \\ |Q_{4f}\rangle &= 0.7071|4\rangle + 0.7071|5\rangle. \end{aligned} \quad (22)$$

Now we build the unitary transformation (U_1) that maps the entry states in the ladder form to the final discriminable configuration. Following the procedure described by equations (13) and (14), we obtain:

$$U_1 = \begin{bmatrix} 0.7071 & 0 & 0.2041-0.2041i & 0.2041+0.2041i & 0.1608+0.2396i & 0.3757-0.3300i & 0 & 0 \\ 0 & 0.7071 & 0.2041+0.2041i & 0.2041-0.2041i & 0.4884-0.1074i & 0.0979+0.2715i & 0 & 0 \\ 0 & 0 & 0.8165 & 0 & -0.1511+0.0977i & -0.5375-0.1097i & 0 & 0 \\ 0 & 0 & 0 & 0.8165 & -0.4981-0.2299i & 0.0639+0.1681i & 0 & 0 \\ -0.3536+0.3536i & -0.3536-0.3536i & 0 & 0.4082 & 0.4981+0.2299i & 0.0639-0.1681i & 0 & 0 \\ -0.3536-0.3536i & -0.3536+0.3536i & 0.4082 & 0 & 0.1511-0.0977i & 0.5375+0.1097i & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}.$$

As the last step, we wish to decompose the resultant matrix $U = U_1 U_0$ as a product of one-qudit rotations. Therefore, we have

$$U = R_{1,2}^\dagger R_{1,3}^\dagger \dots R_{7,8}^\dagger$$

(23)

$$R_{2,5}^\dagger = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 0.8929 & 0 & -0.4239-0.1518i & \dots \\ \vdots & \vdots & \ddots & \vdots & \dots \\ 0 & -0.4239+0.1518i & 0 & -0.8929 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix},$$

where $R_{1,5}^\dagger = R_{1,6}^\dagger = R_{1,7}^\dagger = R_{1,8}^\dagger = R_{2,7}^\dagger = R_{2,8}^\dagger = I$,

$$R_{1,2}^\dagger = \begin{bmatrix} 0.7071 & 0.7071 & 0 & \dots & 0 \\ 0.7071 & -0.7071 & 0 & \dots & 0 \\ 0 & 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix},$$

$$R_{2,6}^\dagger = \begin{bmatrix} 1 & 0 & \dots & 0 & 0 \\ 0 & 0.9594 & \dots & -0.1233+0.2536i & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & -0.1233-0.2536i & \dots & -0.9594 & 0 \\ 0 & 0 & \dots & 0 & 1 \end{bmatrix},$$

$$R_{1,3}^\dagger = \begin{bmatrix} 0.8165 & 0 & 0.5774 & \dots & 0 \\ 0 & 1 & 0 & \dots & 0 \\ 0.5774 & 0 & -0.8165 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 \end{bmatrix},$$

$$R_{3,4}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & -0.6776-0.4981i & -0.5333+0.0914i & \dots & \dots \\ \dots & -0.5333-0.0914i & 0.6776-0.4981i & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix},$$

$$R_{1,4}^\dagger = \begin{bmatrix} 0.8660 & 0 & 0 & 0.5 & \dots & 0 \\ 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ 0.5 & 0 & 0 & -0.8660 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & 0 & \dots & 1 \end{bmatrix},$$

$$R_{3,5}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0.4375 & 0 & -0.3023+0.8469i & \dots \\ \dots & 0 & 1 & 0 & \dots \\ \dots & -0.3023-0.8469i & 0 & -0.4375 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix},$$

$$R_{2,3}^\dagger = \begin{bmatrix} 1 & 0 & 0 & \dots \\ 0 & 0.7224-0.3407i & -0.5392-0.2671i & \dots \\ 0 & -0.5392+0.2671i & -0.7224-0.3407i & \dots \\ \dots & \dots & \dots & \dots \end{bmatrix},$$

$$R_{3,6}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0.8262 & \dots & 0.0453-0.5615i & \dots \\ \dots & \vdots & \ddots & \vdots & \dots \\ \dots & 0.0453+0.5615i & \dots & -0.8262 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix},$$

$$R_{2,4}^\dagger = \begin{bmatrix} 1 & 0 & 0 & 0 & \dots \\ 0 & 0.6865 & 0 & -0.5482+0.4777i & \dots \\ 0 & 0 & 1 & 0 & \dots \\ 0 & -0.5482-0.4777i & 0 & -0.6865 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix},$$

$$R_{3,7}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & \dots & 0 & 0 \\ \dots & 0.9727 & \dots & -0.2320 & 0 \\ \dots & \vdots & \ddots & \vdots & \vdots \\ \dots & -0.2320 & \dots & -0.9727 & 0 \\ \dots & 0 & \dots & 0 & 1 \end{bmatrix},$$

$$R_{3,8}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & \dots & 0 & 0 \\ \dots & 0.7485 & \dots & 0 & 0.6631 \\ \dots & \vdots & \ddots & \vdots & \vdots \\ \dots & 0 & \dots & 1 & 0 \\ \dots & 0.6631 & \dots & 0 & -0.7485 \end{bmatrix},$$

$$R_{4,5}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0 & \dots & -0.6608+0.7506i & \dots \\ \dots & -0.6608-0.7506i & \dots & 0 & \dots \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix},$$

$$R_{4,6}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0.0240 & 0 & -0.9996-0.0164i & 0 \\ \dots & 0 & 1 & 0 & 0 \\ \dots & -0.9996+0.0164i & 0 & -0.0240 & 0 \\ \dots & 0 & 0 & 0 & 1 \end{bmatrix},$$

$$R_{4,7}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0.8438 & \dots & 0.5367 & 0 \\ \dots & \dots & \ddots & \dots & \dots \\ \dots & 0.5367 & \dots & -0.8438 & 0 \\ \dots & 0 & \dots & 0 & 1 \end{bmatrix},$$

$$R_{4,8}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0.9255 & \dots & 0 & 0.3788 \\ \dots & \dots & \ddots & \dots & \dots \\ \dots & 0 & \dots & 1 & 0 \\ \dots & 0.3788 & \dots & 0 & -0.9255 \end{bmatrix},$$

$$R_{5,6}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0.0 & \dots & -0.6606-0.7507i & 0 \\ \dots & -0.6606+0.7507i & \dots & 0.0 & 0 \\ \dots & 0 & \dots & 0 & 1 \end{bmatrix},$$

$$R_{5,7}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0.8615 & \dots & 0.5077 & 0 \\ \dots & \dots & \ddots & \dots & \dots \\ \dots & 0.5077 & \dots & -0.8615 & 0 \\ \dots & 0 & \dots & 0 & 1 \end{bmatrix},$$

$$R_{5,8}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 0.2894 & \dots & 0 & -0.9572 \\ \dots & \dots & \ddots & \dots & \dots \\ \dots & 0 & \dots & 1 & 0 \\ \dots & -0.9572 & \dots & 0 & -0.2894 \end{bmatrix},$$

Table I: A specification of the parameters α, β, γ , and δ involved in the sequence of the two-level operations in the example. The angles α, β, γ , and δ are given in degrees.

	α	β	$\gamma/2$	δ
$R_{1,2}$	90.0	0.0	45.0	180.0
$R_{1,3}$	90.0	0.0	35.27	180.0
$R_{1,4}$	90.0	0.0	30.0	180.0
$R_{2,3}$	-90.0	-2.2066	-36.9938	-128.4034
$R_{2,4}$	-90.0	-138.9380	46.6463	-41.0619
$R_{2,5}$	90.0	160.2880	26.76	19.7119
$R_{2,6}$	-90.0	-115.9253	16.3825	-64.0746
$R_{3,4}$	-90.0	-26.6549	32.7541	134.0108
$R_{3,5}$	-90.0	-109.6455	64.0555	-70.3544
$R_{3,6}$	-90.0	-94.6115	-34.2896	-85.3884
$R_{3,7}$	90.0	180.00	13.4187	0.0
$R_{3,8}$	90.0	0.0	41.5393	180.00
$R_{4,5}$	-90.0	-131.3609	90.00	-48.6390
$R_{4,6}$	90.0	179.1897	88.6247	0.8102
$R_{4,7}$	90.0	0.0	32.4564	180.0
$R_{4,8}$	90.0	0.0	22.2561	180.0
$R_{5,6}$	90.0	131.3456	90.0	48.6543
$R_{5,7}$	90.0	0.0	30.5145	180.0
$R_{5,8}$	90.0	180.0	73.1779	0.0
$R_{6,7}$	90.0	0.0	90.0	180.0
$R_{6,8}$	90.0	180.0	45.0	0.0
$R_{7,8}$	26.8469	0.0	90.0	53.6938

$$R_{6,7}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 1 & 0 & 0 & 0 \\ \dots & 0 & 0.0 & 1.00 & 0 \\ \dots & 0 & 1.00 & 0.0 & 0 \\ \dots & 0 & 0 & 0 & 1 \end{bmatrix}.$$

$$R_{6,8}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 1 & 0 & 0 & 0 \\ \dots & 0 & 0.7071 & 0 & -0.7071 \\ \dots & 0 & 0 & 1 & 0 \\ \dots & 0 & -0.7071 & 0 & -0.7071 \end{bmatrix}.$$

$$R_{7,8}^\dagger = \begin{bmatrix} \dots & \dots & \dots & \dots & \dots \\ \dots & 1 & 0 & 0 & 0 \\ \dots & 0 & 1 & 0 & 0 \\ \dots & 0 & 0 & 0.0 & 0.5921-0.8058i \\ \dots & 0 & 0 & 1.00 & 0.0 \end{bmatrix}.$$

Specifying the parameters $(\alpha, \beta, \gamma, \delta)$ in each step of the decomposition we conclude the the unambiguous discrimination protocol. The parameter's values are given in Table I.

V. CONCLUSIONS

We showed a general algorithm to perform the optimal discrimination of N linearly independent non-

orthogonal pure quantum states by means of semidefinite programming and norm minimization. In addition, we presented a simple computational code that takes as input the ensemble of non-orthogonal states and outputs the best set of discriminable states and the sequence of two-level rotation matrices. As a numerical example, we studied an USD attack to the BB84 protocol.

Acknowledgments

We are grateful to Marcello A. Talarico, Fernando G. S. L. Brandão, S. Pádua and C. Saavedra for valuable discussions. This work was supported by FAPEMIG, CNPq, Instituto do Milênio de Informação Quântica.

Appendix A: DETERMINATION OF THE INCONCLUSIVE PROBABILITY AMPLITUDES

In this appendix we supply the procedure to calculate the inconclusive probability amplitudes in the final configuration, i.e, g_i , for $i = N+1$ to $\frac{1}{2}N(N+3)-1$. We detail the procedure for the case $N = 3$, and then discuss the generalization, presenting a MATLAB code for arbitrary N .

Let's consider three non-orthogonal quantum states in a ladder form. It is convenient to factorize a global phase and set the coefficient multiplying the ket $|1\rangle$ as real, for all states, then:

$$\begin{aligned} |Q_1\rangle &= |1\rangle \\ |Q_2\rangle &= a_1 |1\rangle + a_2 e^{i\theta_1} |2\rangle \\ |Q_3\rangle &= a_3 |1\rangle + a_4 e^{i\theta_2} |2\rangle + a_5 e^{i\theta_3} |3\rangle \end{aligned} \quad , \quad (\text{A1})$$

The final discriminable configuration in the extended Hilbert space is

$$\begin{aligned} |Q_{1f}\rangle &= g_1 |1\rangle + g_4 |4\rangle + g_5 |5\rangle \\ |Q_{2f}\rangle &= g_2 |2\rangle + g_6 |4\rangle + g_7 |5\rangle \\ |Q_{3f}\rangle &= g_3 |3\rangle + g_8 |4\rangle \end{aligned} \quad (\text{A2})$$

The conclusive probability amplitudes g_1, g_2, g_3 are determined via SDP. The inconclusive probability amplitudes g_4 to g_8 must be determined preserving normalization and scalar products among the vectors. From normalization we have,

$$|g_1|^2 + |g_4|^2 + |g_5|^2 = 1 \quad (\text{A3})$$

$$|g_2|^2 + |g_6|^2 + |g_7|^2 = 1 \quad (\text{A4})$$

$$|g_3|^2 + |g_8|^2 = 1 \quad , \quad (\text{A5})$$

and the scalar products are

$$\begin{aligned} \langle Q_{1f} | Q_{2f} \rangle &= \langle Q_1 | Q_2 \rangle , \\ \langle Q_{1f} | Q_{3f} \rangle &= \langle Q_1 | Q_3 \rangle , \\ \langle Q_{2f} | Q_{3f} \rangle &= \langle Q_2 | Q_3 \rangle . \end{aligned} \quad (\text{A6})$$

(A6) can be written in the form

$$g_4^* g_6 + g_5^* g_7 = a_1 \quad (\text{A7})$$

$$g_4^* g_8 = a_3 \quad (\text{A8})$$

$$g_6^* g_8 = a_2 a_4 e^{i(\theta_2 - \theta_1)} + a_1 a_3 \quad . \quad (\text{A9})$$

From (A8) it is observed that g_4 and g_8 can be taken as real. Due to (A5) we have $g_8 = +\sqrt{1 - |g_3|^2}$. Substituting g_8 in (A8) we get $g_4 = +a_3/\sqrt{1 - |g_3|^2}$. Substituting g_8 in (A9) we have

$$g_6 = \frac{[a_2 a_4 \cos \Theta + a_1 a_3] - i[a_2 a_4 \sin \Theta]}{\sqrt{1 - |g_3|^2}} \quad . \quad (\text{A10})$$

with $\Theta = \theta_2 - \theta_1$.

g_5 and g_7 are still undetermined. Writing them as $g_5 = g_5^R + i g_5^{Im}$ e $g_7 = g_7^R + i g_7^{Im}$, and using (A3), (A4), (A7), then

$$g_7^R = \pm \sqrt{b - (g_7^{Im})^2}, \quad (\text{A11})$$

$$g_5^R = \pm \sqrt{a - (g_5^{Im})^2}, \quad (\text{A12})$$

$$g_7^R g_5^R + g_5^{Im} g_7^{Im} = c, \quad (\text{A13})$$

$$g_5^R g_7^{Im} - g_5^{Im} g_7^R = d, \quad (\text{A14})$$

with a, b, c and d reals given by

$$\begin{aligned} a &= 1 - |g_1|^2 - |g_4|^2, \\ b &= 1 - |g_2|^2 - |g_6|^2, \\ c &= a_1 - \frac{a_3[a_1 a_3 + a_2 a_4 \cos \Theta]}{1 - |g_3|^2}, \\ d &= \frac{a_3 a_2 a_4 \sin \Theta}{1 - |g_3|^2}. \end{aligned} \quad (\text{A15})$$

Substituting (A11) and (A12) into (A13) and (A14) we have

$$\sqrt{a - (g_5^{Im})^2} \sqrt{b - (g_7^{Im})^2} + g_5^{Im} g_7^{Im} = c, \quad (\text{A16})$$

$$\sqrt{a - (g_5^{Im})^2} g_7^{Im} - g_5^{Im} \sqrt{b - (g_7^{Im})^2} = d, \quad (\text{A17})$$

From (A16) we have

$$g_7^{Im} = \frac{2c g_5^{Im} \pm \sqrt{(2c g_5^{Im})^2 - 4(b(g_5^{Im})^2 + c^2 - ab)a}}{2a} \quad (\text{A18})$$

and from (A17)

$$\begin{aligned} (g_7^{Im})^4 &+ \left[\frac{(4d^2 - 2ab)(g_5^{Im})^2 + 2ad^2}{a} \right] (g_7^{Im})^2 + \\ &+ \left[\frac{b^2(g_5^{Im})^4 - 6d^2 b(g_5^{Im})^2 + d^4}{a} \right] = 0 \end{aligned} \quad (\text{A19})$$

Substituting (A18) into (A19) we find that g_5^{Im} must obey the following polynomial

$$A(g_5^{Im})^8 + B(g_5^{Im})^6 + C(g_5^{Im})^4 + D(g_5^{Im})^2 + E = 0 \quad (A20)$$

with real coefficients

$$\begin{aligned} \mathbf{A} &= [(4d^2 - 4ab)(c^2 - ab) - 4c^2d^2]^2, \\ \mathbf{B} &= [(4d^2 - 4ab)(c^2 - ab) + 4c^2d^2] \times \\ &\quad \times [8a^2b(c^2 - ab - d^2)] - 64c^2d^2(c^2 - ab)(2a^2b), \\ \mathbf{C} &= [a^2d^4 + a^2(c^2 - ab)^2 - 2d^2a^2(c^2 - ab)] \times \\ &\quad \times [(4d^2 - 4ab)(c^2 - ab) + 4c^2d^2] + \\ &\quad + [4a^2b(c^2 - ba - d^2)]^2 + \\ &\quad + 64c^2d^2(c^2 - ab)(d^2 + ab)a^2, \\ \mathbf{D} &= [a^2d^4 + a^2(c^2 - ab)^2 - 2d^2a^2(c^2 - ab)] \times \\ &\quad \times [4a^2b(c^2 - ba - d^2)], \\ \mathbf{E} &= [a^2d^4 + a^2(c^2 - ab)^2 - 2d^2a^2(c^2 - ab)]^2 \end{aligned} \quad (A21)$$

The roots of this polynomial are easily obtained [14]. We choose for g_5^{Im} any real root, such that $0 \leq (g_5^{Im})^2 \leq a$. Once g_5^{Im} is determined, we calculate g_7^{Im} from (A18). Then (A11) and (A12) are used to calculate g_7^R and g_5^R . Now all the parameters in the final configuration are determined.

It is straightforward to extend this procedure for arbitrary N . Summarizing: we determine the conclusive amplitudes, g_1 to g_N , by SDP; then, starting from the last inconclusive amplitude, $g_{[\frac{1}{2}N(N+3)-1]}$, we use the relations for normalization and scalar products and thus determine all the remaining parameters, except for g_{2N-1} and g_{3N-2} , which are obtained by means of the polynomial

$$\begin{aligned} A(g_{2N-1}^{Im})^8 + B(g_{2N-1}^{Im})^6 + C(g_{2N-1}^{Im})^4 + \\ + D(g_{2N-1}^{Im})^2 + E = 0. \end{aligned} \quad (A22)$$

```
% Unambiguos State Discrimination
% Wilson R.M. Rabelo (wilson@unifap.br)
% 09/2005 UFMG Quantum Information Group
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BEGIN INPUT - BB84
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Number of states
N=4;
%Hilbert space dimension
% (at least 2N-1)
dim=8;
%Probability of each state in the
% ensemble
Mi=-1*[0.25; 0.25; 0.25; 0.25];
zero=[1;0];
one=[0;1];
lr=(zero+one)/sqrt(2);
```

```
ll=(zero-one)/sqrt(2);
cr=(zero+i*one)/sqrt(2);
cl=(zero-i*one)/sqrt(2);
lr2=kron(lr,lr);
ll2=kron(ll,ll);
cr2=kron(cr,cr);
cl2=kron(cl,cl);
lr3=kron(lr2,lr);
ll3=kron(ll2,ll);
cr3=kron(cr2,cr);
cl3=kron(cl2,cl);
%Input sates QII= Q_1,...,Q_N
QII=[lr3 ll3 cr3 cl3];
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% END INPUT
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% BEGINNING:
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%N initial states:
disp('N initial states:')
disp(QII)
%Scalar product for N initial states:
```

```
for jL=1:N
    for iL=1:N
        Esc_QII=QII(1:dim,iL)'*QII(1:dim,jL);
        esc_QII(iL,jL)=Esc_QII;
    end
end
```

```
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
% Rewrite the entry states
% in ladder form (Uo)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%for ig=1:2*N-1
for ig=1:dim
    for jg=1:N
        c(ig,jg)=0.0;
    end
end
```

```
c(1,1)=1.0;
for ittt=1:N-1
    for j=ittt+1:N
        Sfat_w=0.0;
        if ittt > 1
            for xxw=1:ittt-1
                Sfat_w=Sfat_w+conj(c(ittt-xxw,ittt))*c(ittt-xxw,j);
            end
        end
        if N==2
            c(ittt,j)=esc_QII(ittt,j);
        c(ittt+1,ittt+1)=sqrt(1-conj(c(ittt,ittt+1))*c(ittt,ittt+1));
        continue
    end
    if N >= 3
        if ittt==1
            c(ittt,j)=esc_QII(ittt,j);
```

```

else
    c(ittt,j)=(esc_QII(ittt,j)-Sfat_w)/c(ittt,j);
end
end
end
Swfat=0.0;
for xo=1:ittt
    Swfat=Swfat+conj(c(xo,ittt+1))*c(xo,ittt+1);
end
c(ittt+1,ittt+1)=sqrt(1-Swfat);
end

Q=c;
format short
AAA=zeros(dim);
for ux=1:N
    AAA=AAA +Q(1:dim,ux)*QII(1:dim,ux)';
end
[V_0,Sigma_0,W_0]=svd(AAA);
disp('The unitary matrix Uo ')
disp('to put states in ladder form :')
Uo=V_0*W_0'
disp('Test Uo, i.e, (Uo*)(Uo)=')
disp(Uo*Uo')
disp('Initial configuration:[Q1...QN]')
disp(' in ladder form:')
for iu=1:N
    Qcc(:,iu)=Uo*QII(1:dim,iu);
end
Q
pause
format long
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%Scalar product for N states
% in ladder form:
for jL=2:N
    for iL=1:jL-1
        Esc_Q=Q(1:N,iL)'*Q(1:N,jL);
        esc_Q(iL,jL)=Esc_Q;
    end
end
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%SDP Approach%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
c=sdpvar(N,1);
p=sdpvar(N,1);
F_0=eye(dim);
Qt=Q*inv(Q'*Q);
D=F_0;
for ib=1:N
    D=D-p(ib)*Qt(1:dim,ib)*Qt(1:dim,ib)';
end
yalmip('info');
F= set(D > 0);
for ikp=1:N
    F=F+set(p(ikp)>0);
end
solvesdp(F,Mi'*p);
P=double(p)

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
for jj=1:N
    g(jj)=sqrt(P(jj));
end
disp('Conclusive amplitudes via SDP :')
g
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
pause
for ii=N+1:((N/2)*(N+3))-1
    g(ii)=0.0;
end
g(((N/2)*(N+3))-1)=sqrt(1-g(N)^2);

x=1; Sfatg=0;jjj=0;
w=3;
for jL=N:-1:3
    j=((N/2)*(N+3))-x;
    jjj=jjj+1;
    k=jjj;
    for kL=jL-1:-1:1
        if kL > 1
            k=k+1;
            j=j-k;
            if w > 3
                Sfatg=0;
                P=0;
                fatg=0;
                for f=((w/2)*(w-5))+5:((w/2)*(w-3))+1
                    P=P+1;
                    E(f)=g(((N/2)*(N+3))-f)/g(((N/2)*...
                        (N+3))-x);
                    fatg=conj(g(j-P))*E(f);
                    Sfatg=Sfatg+fatg;
                end
            end
            g(j)=((esc_Q(kL,jL))/(g(((N/2)*...
                (N+3))-x)))-Sfatg;
            g(j)=conj(g(j));
        else
            j=j-k;
            if w > 3
                Sfatg=0;
                P=0;
                fatg=0;
                for f=((w/2)*(w-5))+5:((w/2)*(w-3))+1
                    P=P+1;
                    E(f)=g(((N/2)*(N+3))-f)/g(((N/2)*...
                        (N+3))-x);
                    fatg=conj(g(j-P))*E(f);
                    Sfatg=Sfatg+fatg;
                end
            end
            g(j)=((esc_Q(kL,jL))/(g(((N/2)*...
                (N+3))-x)))-Sfatg;

```

```

        g(j)=conj(g(j));
    end
end
if N==3
    continue
end
if jL > 3
    w=w+1;
    fat=0;
    Sfat1=1-g(jL-1)*conj(g(jL-1));
    x=((w/2)*(w-5))+4;
    for t=((w/2)*(w-5))+5:((w/2)*(w-3))+1
        fat=fat+conj(g(((N/2)*(N+3))-t))*...
            g(((N/2)*(N+3))-t);
    end
    g(((N/2)*(N+3))-x)=sqrt(Sfat1-fat);
else
    continue
end
end
fatp=0.0;
for iik=N+1:2*N-2
    fatp=fatp+g(iik)*conj(g(iik));
end
a=1-g(1)^2-fatp;

fatpp=0.0;
for iiik=2*N:3*N-3
    fatpp=fatpp+g(iiik)*conj(g(iiik));
end
b=1-g(2)^2-fatpp;

fatig=0.0;
for pp=2*N:3*N-3
    fatig=fatig+conj(g(pp+(1-N)))*g(pp);
end
g2N_1_g3N_2=esc_Q(1,2)-fatig;
c=real(g2N_1_g3N_2);
d=imag(g2N_1_g3N_2);

if d==0.0
    g(2*N-1)=sqrt(a);
    g(3*N-2)=sqrt(b);
    format short;

elseif( (c < 1.0e-010) & (c > -0.1e-05))
    g(3*N-2)=sqrt(b);
    g_2N_1_im=-sqrt(a);
    g(2*N-1)=complex(0,g_2N_1_im);
    format short;

elseif ((d < 1.0e-010) & (d > -0.1e-05))
    g(2*N-1)=sqrt(a);
    g(3*N-2)=sqrt(b);
    format short;

else
    %%%% Polynomial%%%%
    AA=[(4*d^2-4*a*b)*(c^2-a*b)-4*c^2*d^2]^2;

    BB=[(4*d^2-4*a*b)*(c^2-a*b)+4*c^2*d^2]*...
        [8*a^2*b*(c^2-a*b-d^2)]-64*c^2*d^2*...
        (c^2-a*b)*(2*a^2*b);

    CC=[a^2*d^4+a^2*((c^2-a*b)^2)-2*d^2*a^2*...
        (c^2-a*b)]*[(4*d^2-4*a*b)*(c^2-a*b)+...
        4*c^2*d^2]+[4*a^2*b*(c^2-a*b-d^2)]^2+...
        64*c^2*d^2*(c^2-a*b)*(d^2+a*b)*a^2;

    DD=[a^2*d^4+a^2*((c^2-a*b)^2)-2*a^2*...
        d^2*(c^2-a*b)]*4*a^2*b*(c^2-a*b-d^2);

    EE=[a^2*d^4+a^2*(c^2-a*b)^2-2*...
        a^2*d^2*(c^2-a*b)]^2;
    %%%%%%%%%%%%%%%
    Pii=[AA 0 BB 0 CC 0 DD 0 EE];
    rr=roots(Pii);

    for ih=1:8
        if isreal(rr(ih))
            if (rr(ih) < 1.0) & (rr(ih) > 0.0)
                g_2N_1_im=rr(ih);
                g3N_2_im=[2*c*g_2N_1_im+...
                    sqrt((2*c*g_2N_1_im)^2-4*a*...
                    [b*g_2N_1_im^2+c^2-a*b]))]/(2*a);
                g_2N_1_R=-sqrt(a-g_2N_1_im^2);
                g3N_2_R=-sqrt(b-g3N_2_im^2);
                %%%%%%%%%%%%%%%
                ccc=g3N_2_R*g_2N_1_R+...
                    g3N_2_im*g_2N_1_im;
                ddd=g_2N_1_R*g3N_2_im-...
                    g_2N_1_im*g3N_2_R;
                ai=g_2N_1_R;
                bi=g_2N_1_im;
                g(2*N-1)=complex(ai,bi);
                aii=g3N_2_R;
                bii=g3N_2_im;
                g(3*N-2)=complex(aii,bii);
                format short;
            else
                continue
            end
        else
            if ih==8
                disp('roots=')
                disp(rr)
                disp('Polynomial roots ')
                disp(' are not real!')
                disp('Check input states!')
                disp('Input states cannot')
                disp(' be linearly dependent.')
                return
            else
                continue
            end
        end
    end
end

```

```

        end
    end
end
disp(' ')
disp('Conclusive and inconclusive')
disp(' amplitude probabilities :')
    g
    pause

%%%Building final vectors Qf %%%%
if dim==2*N-1;
    xdim=2*N-1;
else
    xdim=dim;
end

for uk=1:N;
    for ui=1:xdim;
        Qf(ui,uk)=0.0;
    end
end
%-----
jj=0 ;
for iiiv=1:N;

    jj=jj+1;
    for uk=1:N;
        if uk==jj;
            Qf(uk,iiiv)=g(uk);
        else uk<N+1 ;
            continue
        end
    end

    end
end
S=N;
uk=0;
iyy=2*N-1;
for j=1:N;
    uk=uk+1;
    if j==1 ;
        for zk=S+1:S+(N-j);
            Qf(zk,uk)=g(zk) ;
        end

        elseif j<= N-1 ;
            xx=uk-2;
            for zk=S+1:S+(N-j)+1;
                zmim=S+1;
                Qf(iyy-(N-uk)+(zk-zmim)-...
                    xx,uk)=g(zk);
            end
        else
            zk=((N/2)*(N+3))-1;
            Qf(N+1,uk)=g(zk);
        end
    end
    S=zk;

```

```

end
    Qf
    pause

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
A=zeros(xdim);
for ux=1:N
    A=A +Qf(1:xdim,ux)*Q(1:xdim,ux)';
end
disp('A=')
disp(A)
[V,Sigma,W]=svd(A) ;
disp('The unitary matrix U1 :')
U1=V*W'
disp('Test U1, i.e. (U1*)(U1)=')
disp(U1*U1')
disp('The final configuration:')
disp(' [Q1f Q2f Q3f ... QNf]')
    for iu=1:N
        Qfc(:,iu)=U1*Q(1:xdim,iu);
    end
    disp(Qfc)

disp('decomposing (U1)*(Uo)')
disp('U=U1*Uo')
disp(U1*Uo)
pause

%%%Decomposing U=(U1)(Uo)%%%%%%%%%
U_aux=U1*Uo;
k=0;
for ifg=1:xdim-2 ;
    for j=ifg+1:xdim ;
        k=k+1;
        a=U_aux(ifg,ifg);
        b=U_aux(j,ifg);
        c=sqrt(conj(a)*a+conj(b)*b);
        a=a/c;
        b=b/c;
        if ((conj(b)*b)<0.0001) & (j==ifg+1);
            V=eye(xdim);
            R(:, :, ifg,j)=V;
            U_aux=V*U_aux;
            continue
        elseif ((conj(b)*b)<0.0001) & (j>ifg+1);
            V=eye(xdim);
            V(ifg,ifg)=conj(a);
            R(:, :, ifg,j)=V ;
            U_aux=V*U_aux;
            continue
        else ((conj(b)*b)>0.00000001);
            V=eye(xdim);
            V(ifg,ifg)=conj(a);
            V(ifg,j)=conj(b);
            V(j,ifg)=b;
            V(j,j)=-a;

```

```

        R(:, :, ifg, j) = V ;
        U_aux = V * U_aux;
        continue
    end
end
end
V_aux = U_aux;
ifg = xdim - 1;
j = ifg + 1;
k = k + 1;
V = eye(xdim);
V(ifg, ifg) = conj(V_aux(ifg, ifg));
V(ifg, j) = conj(V_aux(j, ifg));
V(j, ifg) = conj(V_aux(ifg, j));
V(j, j) = conj(V_aux(j, j));
R(:, :, ifg, j) = V;
for ifg = 1:xdim - 1 ;
    for j = ifg + 1:xdim ;
        if j == ifg ;
            continue
        else
            Rotation(:, :, ifg, j) = R(:, :, ifg, j)';
            disp('Rotation'), disp([ifg j])
            disp(Rotation(:, :, ifg, j))
        end
    end
end
pause
end
disp('Test Rotations')
disp('R_(d-1,d)R_(d-2,d)...R(1,3)R_(1,2)U=')
U_final = V * V_aux;
disp(U_final)
%%% END %%%

```

-
- [1] A. Peres, *Quantum Theory: Concepts and Methods* (Kluwer Academic Publishers, Dordrecht, 1993).
- [2] C. H. Bennett, Phys. Rev. Lett. A **68**, 3121 (1992).
- [3] I. D. Ivanovic, Phys. Lett. A **123**, 257 (1987).
- [4] D. Dieks, Phys. Lett. A **126**, 303 (1988).
- [5] A. Peres, Phys. Lett. A **128**, 19 (1988).
- [6] G. Jaeger, A. Shimony, Phys. Lett. A **197**, 83 (1995).
- [7] A. Chefles, Phys. Lett. A **239**, 339 (1998).
- [8] Y. C. Eldar, IEEE Trans. Inform. Theory, **49**, 446-456 (2003).
- [9] Sonja Franke-Arnold, E. Andersson, S. M. Barnett, S. Stenholm, Phys. Rev. A **63**, 052301 (2001).
- [10] Y. Sun, M. Hillery, J. A. Bergou, Phys. Rev. A **64** 022311 (2001).
- [11] L. Roa, J. C. Retamal, C. Saavedra, Phys. Rev. A **66**, 012103 (2002).
- [12] S. Boyd, L. Vandenbergue, "Semidefinite Programming", Siam Review, March 1996.
- [13] M. A. Neumark, Izv. Akad. Nauk. SSSR, Ser. Mat. **4**, 53 277 (1940).
- [14] R. A. Horn, C. Johnson, *Matrix Analysis*, (Cambridge University Press, 1991).
- [15] M. Reck, A. Zeilinger, et. al., Phys. Rev. Lett. **73** , 58 (1994).
- [16] M. Dusek, M. Jahma, N. Lütkenhaus, Phys. Rev. A **62**, 022306 (2000).
- [17] S. J van Enk, Phys. Rev. A **66**, 042313 (2002).
- [18] C.H. Bennett and G. Brassard, in *Proceedings of the IEEE International Conference on Computer, Systems, and Signal Processing, Bangalore, India* (IEEE, New York, 1984), pp. 175-179.